

RCS Based Hardware-in-the-loop Intelligent System Design and Performance Measurement

**Ananthakrishnan S.,
Chief Engineer,
Pathway Tech. Inc.,
Blue Bell, PA**

**Agrawal, Sunil,
Professor,
Univ. of Delaware,
Newark, DE**

**Venugopal, Ravinder,
Consultant,
Pathway Tech. Inc.,
Blue Bell, PA**

**Demeri, Mahmoud
Senior Tech. Specialist
Ford Research Lab
Dearborn, MI**

Abstract:

In this paper, we present an approach to the design of intelligent systems based on RCS architecture that allows seamless transition from modeling and non real-time simulation to real-time simulation and subsequent hardware-in-the-loop testing. This methodology provides a unified, structured, hierarchical environment, so that “analytical design” of intelligence can be seamlessly transferred to machine/manufacturing process intelligence. As part of the research, we present two case studies wherein we demonstrate how RCS architecture and functionality can be incorporated using commercial software and hardware environment. The enhancements to commercial software in the areas of (i) knowledge hierarchy, (ii) open, modular, and structured programming using RCS architecture, (iii) minimal software programming, (iv) advanced control design methodologies, and (v) efficient numerical schemes for optimization provide a framework for comparing qualitative and quantitative measures of performance improvement over traditional industrial automation hardware that uses PID cards, programmable logic controllers (PLCs), and other microprocessor based controllers with limited functionality.

In order to compete in the global market place, engineering organizations are under increasing pressure to design, develop, and deploy products in the market place as quickly as possible with first time quality. In order to achieve these objectives, it is necessary to streamline the design and development process, namely, “transfer of analytical design of intelligence to mechatronics intelligence” in an efficient and expedient manner. Using Real-time Control System (RCS) architecture that organizes the elements of intelligence to create functional relationships and information flow across levels following principles of hierarchy and assigned responsibilities at each level [1, 2], we have implemented control systems for two applications, namely, a cable robot, and electrohydraulic test system.

The objective of this research effort was to conduct case studies on how RCS architecture can be used in a flexible automation scenario where traditional industrial control cards (hardware) do not provide adequate measures of performance. In addition, industrial control hardware and real-time software primarily focus on embedded software with no structured approach or methodologies for real-time simulation or hardware testing of intelligent system design. Further, in commercial software used for hardware-in-the-loop testing, there is a general lack of well-defined intelligent infrastructure. Hence, from the standpoint of intelligent system design and performance metrics, through these design examples, we demonstrate the need for a unified environment for design and development of intelligent systems that combine knowledge hierarchy, computational schemes, dynamic models, etc., so that platform configuration, and repetitive coding can be minimized [3].

Intelligent control has been a focus of attention for researchers over the past three decades. Initially, it was viewed as interaction of artificial intelligence and control systems [4]. Another major attempt to formalize the discipline of intelligent controls includes theories of nested hierarchical information structures to address control of complex systems [2, 5, 6]. Nonetheless, all these approaches emphasize the importance of “analytical design” of intelligent machines and focus of system functions pertaining to machine intelligence.

In summary, design of intelligent systems based on methodologies described in [2, 4, 5, 6] have the following three common characteristics: (i) utilization and implementation of concepts and ideas from diverse disciplines, (ii) “additional controllers” to accommodate intelligent system performance that utilize knowledge based techniques to meet performance requirements, (iii) emphasis on the overall system coordination and integration as opposed to control specific system components. In this paper, through the two case studies we demonstrate how commercial software can be adapted to meet these objectives of intelligent system design through hardware-in-the-loop testing. The qualitative measures of performance enhancement with this design approach are, (i) structured environment using RCS architecture (ii) system models for rapid plug and play design, (iii) minimal platform configuration and coding. Quantitative measures of performance enhancement are, (i) improved control design and implementation techniques over commercial control hardware, (ii) seamless migration from simulation to hardware testing, (iii) cost effective intelligent system design modules for commercial environment.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE AUG 2002		2. REPORT TYPE		3. DATES COVERED 00-00-2002 to 00-00-2002	
4. TITLE AND SUBTITLE RCS Based Hardware-in-the-loop Intelligent System Design and Performance Measurement				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Pathway Tech. Inc,Blue Bell,PA,19422				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop (PerMIS -02), Gaithersburg, MD on August 13-15, 2002					
14. ABSTRACT see report					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 10	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

System Dynamics and Control

Cable Robot Dynamics and Control:

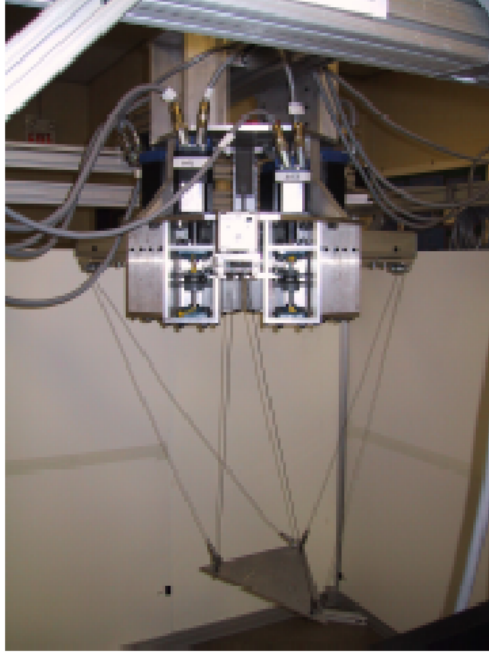


Figure 3. Cable Robotic System

In order to systematically explore the questions of configuration design, coordination, and intelligent control, we have implemented a hardware-in-the-loop control design environment for a cable robot as shown in Figure 3. The system consists of 6 degrees-of-freedom cable robot mounted on a two degree-of-freedom X-Y gantry structure. Cable suspended robots have one unique property – *they carry loads in tension but not in compression*. Due to this feature, well-known results in robotics for trajectory planning and control are not directly applicable to cable robots, but must be modified to reflect the constraints of positive cable tensions [8, 9]. In this paper, we present RCS based control implementation of design algorithms taking into consideration that the cables have to be in tension for effective control.

Based on Newton-Euler formulation, the equations of motion for the cable system without considering the gantry motion can be written as

$$\begin{bmatrix} m\ddot{x}_m \\ m\ddot{y}_m \\ m\ddot{z}_m - g \\ I \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} + \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{pmatrix} \times I \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{pmatrix} \end{bmatrix} = -\tilde{\mathbf{J}}^T(q)u \quad (i)$$

where in equation (i), m is the mass matrix, I is the moment of inertia of the end-effector about its center of mass with respect to the basis vector $[b_1 \ b_2 \ b_3]$ [8]. The above equation can be rewritten in the following form

$$D(x)\ddot{x} + C(x, \dot{x})\dot{x} + G(x) = -\tilde{\mathbf{J}}^T(q)u \quad (ii)$$

where $x = [x_m \ y_m \ z_m \ \mathbf{y} \ \mathbf{q} \ \mathbf{f}]^T$ and it's derivatives refer to the configuration of the end-effector plate and q is a vector of cable lengths. The functional relation between $\mathbf{w}_i, \mathbf{a}_i$ and $(\mathbf{y}, \mathbf{q}, \mathbf{f})$ is given by

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S\mathbf{q} \\ 0 & C\mathbf{y} & S\mathbf{y}C\mathbf{q} \\ 0 & -S\mathbf{y} & C\mathbf{y}C\mathbf{q} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{f}} \end{bmatrix} \quad (\text{iii})$$

with $\mathbf{a}_i = \dot{\mathbf{w}}_i$ and $G(x) = [0 \ 0 \ -mg \ 0 \ 0 \ 0]^T$.

Based on the dynamic equations described in equation (ii), a *Lyapunov based controller* was developed with the candidate Lyapunov function in equation (iv)

$$V(\tilde{x}, x) = \frac{1}{2} \dot{x}^T D(x) \dot{x} + \frac{1}{2} \tilde{x}^T K_p \tilde{x} \quad (\text{iv})$$

where $\tilde{x} = x - x^d$ with x^d as the reference trajectory. In order to ensure asymptotic stability of equation (ii) about x^d , a control law was developed in [8], to ensure positive control. The control law given in equation (v)

$$\begin{bmatrix} -\tilde{J}^T & \tilde{X} & \dot{\tilde{X}} \end{bmatrix} \begin{bmatrix} u \\ K_p \\ K_d \end{bmatrix} = G(x) \quad (\text{v})$$

has the structure of $Ay = b$ such that $y > 0$ with \tilde{X} having a dimension of (6 X 6) with diagonal entries of \tilde{x}_i . The resulting optimization problem to ensure positive control results in minimizing $\|Ay - b\|_2$ and was solved using *lsqnonneg*, a nonnegative least squares problem solver. The computations of pseudo-inverse and Singular Value Decomposition (SVD) are directly applicable to least squares optimization problems. The technique of *lsqnonneg* is an iterative process. Consider a system of equations represented by $Ay = b$, the technique consists of computing the residue $R = b - Ay_e$, where y_e is the initial guess or estimate of the solution. The main process of *lsqnonneg* consists of iteratively computing $w = A^T R$, and improving the estimates y_e in each solution step. The process iterates till all elements of w are less than the specified tolerance, and the number of iterations reach an upper limit. In each iteration step, the pseudo-inverse of a subset of the elements of matrix A is used to compute the new estimates y_e . The pseudo-inverse is computed using SVD. As part of the knowledge hierarchy, the intelligent system design involved developing a real-time S-function C-code API structure in Matlab/Simulink that would allow simultaneous optimized gain selection using *lsqnonneg* and Lyapunov based controller. This was implemented using RCS methodology in the cable robotic system at University of Delaware. The details of RCS based implementation of this design and the results of this experimental study are discussed later.

Electrohydraulic Test System:

New tooling concepts and an advanced binder control unit with individually controlled hydraulic cylinders has recently been developed to allow the local control of metal flow into the die cavity during a stamping operation. Forces are applied on the sheet metal blank using a set of hydraulic cylinders mounted on the lower bolster of the press. In a hydraulic press, the ram depresses the piston of each one of the hydraulic cylinders in the binder area, thus compressing the hydraulic fluid and raising the pressure inside the cylinders. This pressure is transferred through the piston back up to the blank. To obtain the desired force on the blank, the pressure within the hydraulic cylinder is regulated by modulating the flow of hydraulic fluid out of the cylinder by means of a closed-loop control system. But, unlike a hydraulic press where the ram maintains a constant velocity profile, the piston velocity of the mechanical press is a nonlinear function of time, and therefore, the differential equation that relates rate of pressure increase to servo-valve opening is nonlinear. Hence, control of pressure within the cylinder cannot be achieved with simple PID control in the case of a mechanical press, although this would be possible in the case of a hydraulic press for which the piston velocity is constant during the stamping cycle. In this case, standard off-the-shelf PID cards were not able to control the cylinder pressure in a mechanical press.

In this case, we demonstrate an RCS based nonlinear controller [10] using a single cylinder test stand as shown in Figure 4. As can be seen, the hydraulic cylinder in the middle of the platform will be controlled as the table is actuated

using the crank assembly using motors shown in Figure 4. The schematic of the hydraulic system consisting of reservoir, servovalve, and other hydraulic components is shown in Figure 5.

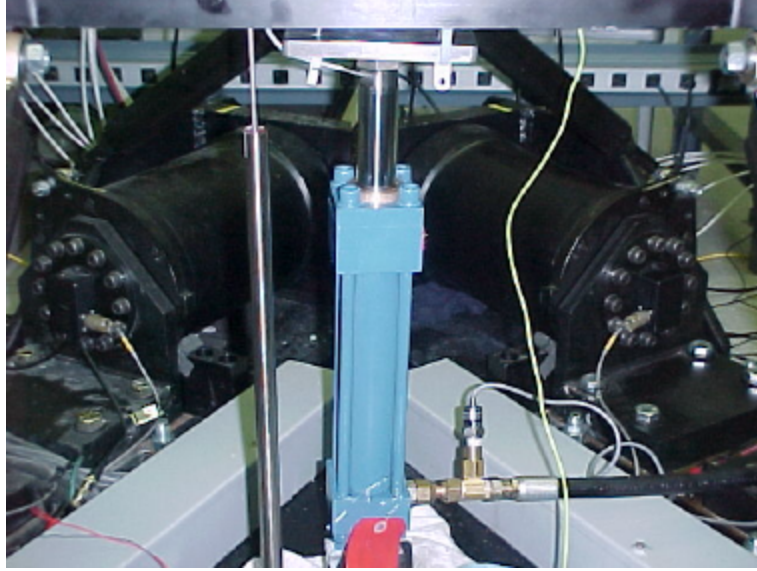


Figure 4. Hydraulic System Test Stand for Single Cylinder Test

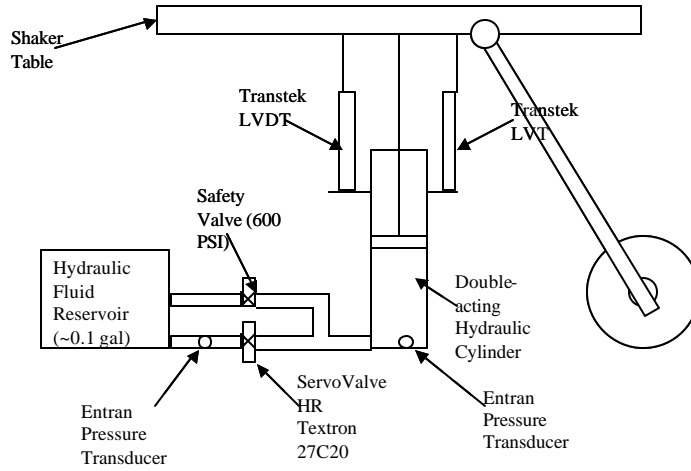


Figure 5. Schematic of the Hydraulic Test System for Closed Loop Control

System Dynamics:

Figure 6 shows the mechanical ram and the hydraulic force control unit in a typical mechanical press. The governing equations of motion for the system shown in Figure 6 consists of cylinder dynamics of the hydraulic system and the dynamics of the mechanical crank drive as described in the following equations (1) and (2):

$$\dot{P}(t) = \frac{b}{A(s - d(t) + e)} \left(A\dot{d}(t) - a(t)K_{servo} \sqrt{\frac{2(P(t) - P_t)}{r}} \right) \quad (1)$$

where $P(t)$ is the cylinder pressure, P_t is the tank pressure, A is the cross-sectional area of the cylinder, $d(t)$ is the displacement of the piston from top-dead-center (TDC), $0 \leq a(t) \leq 1$ is the amount by which the servo-valve is

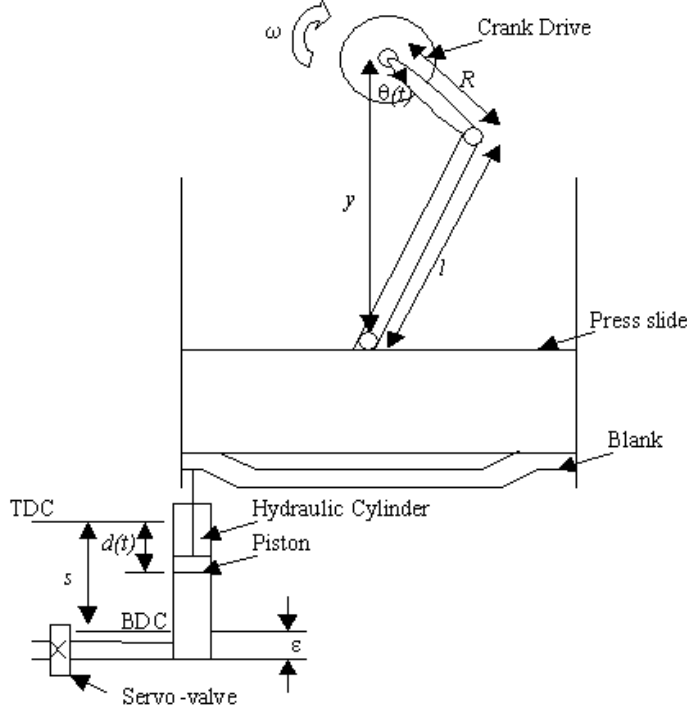


Figure 6. Schematic of Hydraulic Force Actuation in Mechanical Press

opened with $\mathbf{a}(t) = 0$ representing the valve fully closed and $\mathbf{a}(t) = 1$, \mathbf{r} is the density of the hydraulic fluid, K_{servo} is the effective cross-sectional area of the valve orifice, s is the stroke length of the piston and \mathbf{e} is the height of fluid in the cylinder when the piston reaches bottom-dead-center (BDC) as shown in Figure 6. We now mathematically model a simple mechanical crank press drive to obtain a relationship between the displacement of the piston and its velocity, assuming that all the links is rigid as shown in equation (2)

$$d(t) = R \cos \mathbf{q}(t) - \sqrt{l^2 - R^2 \sin^2 \mathbf{q}(t)} - R - l + s. \quad (2)$$

where $\mathbf{q}(t)$ is the crank angle, R is the crank radius, and l is the length of the coupler, as shown in Figure 7. Differentiating Equation (2) results in Equation (3) as shown below.

$$\dot{d}(t) = -R \sin \mathbf{q}(t) \mathbf{w} - \frac{R^2 \sin 2\mathbf{q}(t) \mathbf{w}}{2\sqrt{l^2 - R^2 \sin^2 \mathbf{q}(t)}}, \quad (3)$$

The resulting nonlinear model of equation (1) and (3) describes the dynamics of the hydraulic actuation system used in a mechanical press. The nonlinear nature of the equations clearly indicates that standard PID control will not be sufficient for precise closed-loop pressure control.

Control System Design:

The controller design for the hydraulic force actuation unit is based upon equation (1) and (3) of the preceding section. The nonlinear control technique used for designing the controller is feedback linearization. Feedback linearization is used in the control of nonlinear systems in which the nonlinearity is known and invertible. It involves the use of additional terms in the control signal to cancel out the nonlinearity, after which a classical linear controller is used on the effectively linear system. Using the structure of the nonlinearity, we choose

$$\mathbf{a}(t) = \frac{A}{K_{servo}} \sqrt{\frac{\mathbf{r}}{2(P(t) - P_i)}} \left(\dot{d}(t) - \frac{(s - d(t) + \mathbf{e})}{\mathbf{b}} v(t) \right) \quad (4)$$

where $v(t)$ is a linear control term to be determined subsequently. Substituting equation (1) into equation (4) results in

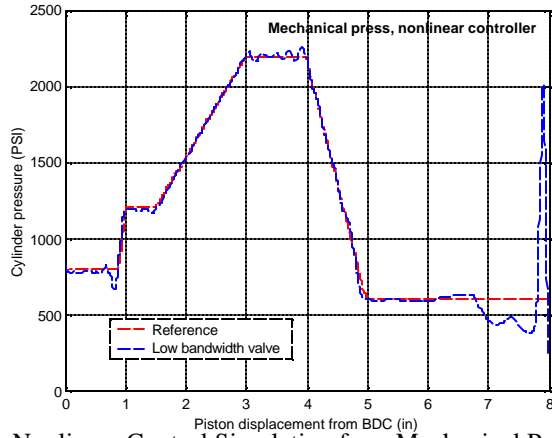
$$\dot{P}(t) = v(t) \quad (5)$$

Thus, the control law in equation (4) converts the nonlinear control problem in equation (1) into a first order linear ODE involving the new control variable $v(t)$. The first type of controller that we choose for the system given by (5) is a PI controller of the form shown in equation (6), namely,

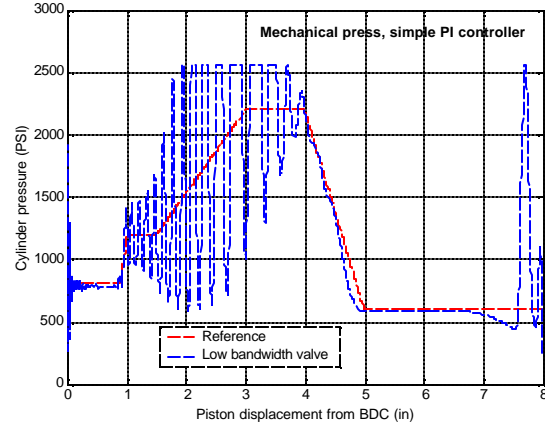
$$v(t) = K_p (P_{ref}(t) - P(t)) + K_i \int_{t_{0i}}^t (P_{ref}(s) - P(s)) ds, \quad (6)$$

where K_p is the proportional gain, $P_{ref}(t)$ is the desired pressure command, and K_i is the integral gain. Substitution of equation (6) into equation (4) results in the following nonlinear control law,

$$\mathbf{a}(t) = \frac{A}{K_{servo}} \sqrt{\frac{\mathbf{r}}{2(P(t) - P_t)}} \left(\dot{d}(t) - \frac{(s - d(t) + \mathbf{e})}{\mathbf{b}} (K_p (P_{ref}(t) - P(t)) + K_i \int_{t_{0i}}^t (P_{ref}(s) - P(s)) ds) \right) \quad (7)$$



Nonlinear Control Simulation for a Mechanical Press



PI Control Simulation for a Mechanical Press

Figure 7. Cylinder Pressure Comparison between Linear and Nonlinear Control

Figure 7 shows the comparison between pressure profiles using linear and nonlinear controllers. The pressure profiles are given from bottom dead center (BDC) of the cylinder, and so must be viewed from right to left. As the ram comes down and makes contact with the cylinder, the initial pressure spike to the right is noticed. Then, as the ram plunges the cylinder, the nonlinear control is able to track the reference pressure trajectory, while the PI control causes significant oscillations. So, a simple change from hydraulic ram to a mechanical ram rendered the conventional control cards inadequate for effective control. The structure of nonlinear control used in this design can be generalized to a class of systems. Essentially, since the structure of nonlinearity is known in pressure control of hydraulic systems, we can use feedback linearization techniques.

A second approach to control of the cylinder pressure in the binder force control unit use estimates of plant dynamics in a pole assignment controller leading to pole assignment adaptive control. A deterministic auto regressive moving average (DARMA) model of the plant represented by equation (8) represents the plant dynamics

$$A(q^{-1})y(t) = q^{-d} B'(q^{-1})u(t) \quad (8)$$

where the coefficients of the polynomial $A(q^{-1}), B'(q^{-1})$ are given by equation (9) and (10)

$$A(q^{-1}) = a_0 + a_1 q^{-1} + \dots + a_{n1} q^{-n1} \quad (9)$$

$$B(q^{-1}) = (b_0 + b_1 q^{-1} + \dots + b_{m_1} q^{-m_1}) q^{-d} \quad (10)$$

The plant dynamics can be estimated based on past measurements of $y(t), u(t)$ by rewriting equation (8) in normalized form where $a_0 = 1$ as follows:

$$y(t) = \mathbf{f}(t-1)^T \mathbf{q} \quad (11)$$

with

$$\mathbf{f}(t-1)^T = [-y(t-1), -y(t-2), \dots, u(t-1), u(t-2) \dots]$$

$$\mathbf{q} = [a_1, a_2, \dots, b_1, b_2, \dots]$$

The parameters of the system can be estimated using recursive least squares (RLS) algorithm and its variation such as RLS with covariance resetting as follows:

$$\hat{\mathbf{q}}(t) = \hat{\mathbf{q}}(t-1) + \frac{P(t-2)\mathbf{f}(t-1)}{1 + \mathbf{f}(t-1)^T P(t-2)\mathbf{f}(t-1)} (y(t) - \mathbf{f}(t-1)^T \hat{\mathbf{q}}(t-1)) \quad (12)$$

$$P(t-1) = P(t-2) - \frac{P(t-2)\mathbf{f}(t-1)\mathbf{f}(t-1)^T P(t-2)}{1 + \mathbf{f}(t-1)^T P(t-2)\mathbf{f}(t-1)} \quad (13)$$

For covariance resetting, equation (13) is reset periodically by a known value of covariance as follows

$$P(t_i - 1) = k_i I \quad (14)$$

The control input $u(t)$ in pole assignment controller is determined by solving the following equation

$$\hat{A}(t, q^{-1})\hat{L}(t, q^{-1}) + \hat{B}(t, q^{-1})\hat{P}(t, q^{-1}) = A^*(q^{-1}) \quad (15)$$

where $\hat{\mathbf{q}}(t)$ consists of estimates $\hat{A}(t, q^{-1}), \hat{B}(t, q^{-1})$ and $\hat{L}(t, q^{-1}), \hat{P}(t, q^{-1})$ are unique polynomials of order $(k-1)$ and $A^*(q^{-1})$ is the desired polynomial selected based on intended closed loop behavior. The feedback control law that combines the estimates of the plant dynamics and equation (15) is given by [12, 13]

$$\hat{L}(t, q^{-1})u(t) = \hat{M}(t)y^*(t) - \hat{P}(t, q^{-1})y(t) \quad (16)$$

where

$$\hat{M}(t) = \frac{1 + a_1^* + \dots}{\hat{b}_1 + \dots} \quad (17)$$

with $y^*(t)$ same as $P_{ref}(t)$, $y(t)$ same as $P(t)$ and $u(t)$ same as the valve command similar to the nonlinear controller.

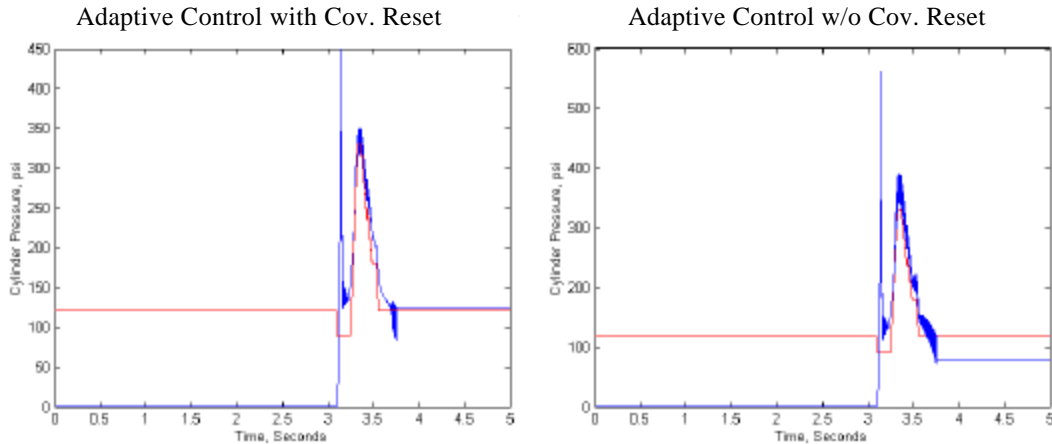


Figure 8. Cylinder Pressure Comparison for Adaptive Controllers

Figure 8 shows the comparison between pressure profiles using adaptive controllers with or without covariance reset. As can be seen, unlike, linear controllers, both the adaptive controllers show a stable response like the nonlinear controller. But, in the case of adaptive controller with covariance resetting, in addition to the tracking, the response is well behaved in the presence of plant and measurement noise, similar to the nonlinear controller. The initial spike in the response is due to the initial transient as the ram comes down and plunges on the cylinder. So, from these analytical results, it is clear that the nonlinearities in the system require linear control structure with adaptation or a nonlinear controller in order to get desired response behavior.

RCS Implementation:

In a goal driven, sensory interactive, and system behavior based intelligent architecture, we have used the four key paradigms and four key elements [1, 2] as guideline for design of intelligent systems. With simulation as the heart of the development process whether it be non real-time or real-time or hardware-in-the-loop, our case studies support the entire intelligent design process without having to transfer data, change design environment, or write extensive custom code. Further, the design process allows models of physical systems, namely, kinematics and dynamics of cable robot and mechanical press test fixture, to be used for non real-time simulation and subsequent hardware-in-the-loop testing of various real-time control schemes, namely, linear, nonlinear, adaptive, and Lyapunov based controllers. In order to avoid repetitive coding we have developed application programmers' interfaces (APIs) that allow optimized integrated code generation, and embedded system options that allow seamless transition of intelligent controllers from simulation to hardware testing.

Figure 9 shows the functional decomposition of the information flow for the cable robotic system. As can be seen from the schematic, the design flow uses the four key paradigms, and four key elements of RCS control node to decompose the design problem into a multi-layered hierarchical control problem. The tension sensor signals {TS1,...,TS6} are fed back to the lowest layer of the hierarchy to close tension feedback control loops using control laws that receive tension request from the Prim process and the filtered sensor data from the sensory processing {SP1, ..., SP6}. Based on whether the tension request is within the desired threshold, the value judgment {VJ1, ..., VJ6} is used to limit the tension request so that appropriate motor commands are generated by the behavior generation module {BJ1, ..., BJ6}. The motor command to the servo drives that actuate the motor are thus assured to be within operational limits in order to achieve effective tension control.

The commands to the *tension control RCS nodes* are generated by the *Prim process*, which also uses the elements of the RCS node to implement the *Lyapunov controller* in order to ensure that positive control is achieved. The optimization problem to ensure positive control results in minimizing $\|Ay - b\|_2$ and was solved using *lsqnonneg*, a nonnegative least squares problem solver. *Since the basic Simulink structure does not support real-time optimization and control, S-function C-code API structure had to be developed as part of the knowledge hierarchy to demonstrate seamless transition from simulation to hardware testing.* As part of this knowledge hierarchy, the computations of pseudo-inverse and Singular Value Decomposition (SVD) are directly applicable to least squares optimization problems and hence the associated algorithms were also developed. As discussed earlier, at each control step the lyapunov controller requires the solution of *lsqnonneg*, which consists of iteratively computing $w = A^T R$, and improving the estimates y_e in each solution step. The process iterates till all elements of w are less than the specified tolerance, and the number of iterations have an upper limit. In each iteration step, the pseudo-inverse of a subset of the elements of matrix A is used to compute the new estimates y_e .

The pseudo-inverse is computed using SVD. The Prim process, based on desired reference trajectories x^d from the E-move process computes the tension request u of equation (v) to the *tension control RCS nodes* based on this Lyapunov controller. In this structure of the Prim process, *lsqnonneg* serves as the VJ module for the *Lyapunov controller* that acts as a BJ module based on desired and actual trajectories that were generated using encoder feedback and associated SP computations that involves computation of *Jacobian* at each control step.

Similar methodologies were used in design of controllers for the electrohydraulic test system used for stamping process automation prototyping and for the sake of brevity not discussed here. In summary, the analytical methodologies, knowledge hierarchy and algorithms developed for estimation, optimization and control of intelligent systems provide a framework for software and associated real-time control development that will allow seamless transition from simulation to hardware. Figure 10 shows the framework of software developed for the case studies. As can be seen, this process can be streamlined for generalized development of knowledge hierarchy and algorithms to allow simulation and hardware testing of

intelligent systems. Figure 11 shows the response plots and GUI used in real-time testing of cable robot. As can be seen, the *Lyapunov controller* works very well while maintaining positive tensions. Similar GUI was developed for design and testing of the electrohydraulic test system.

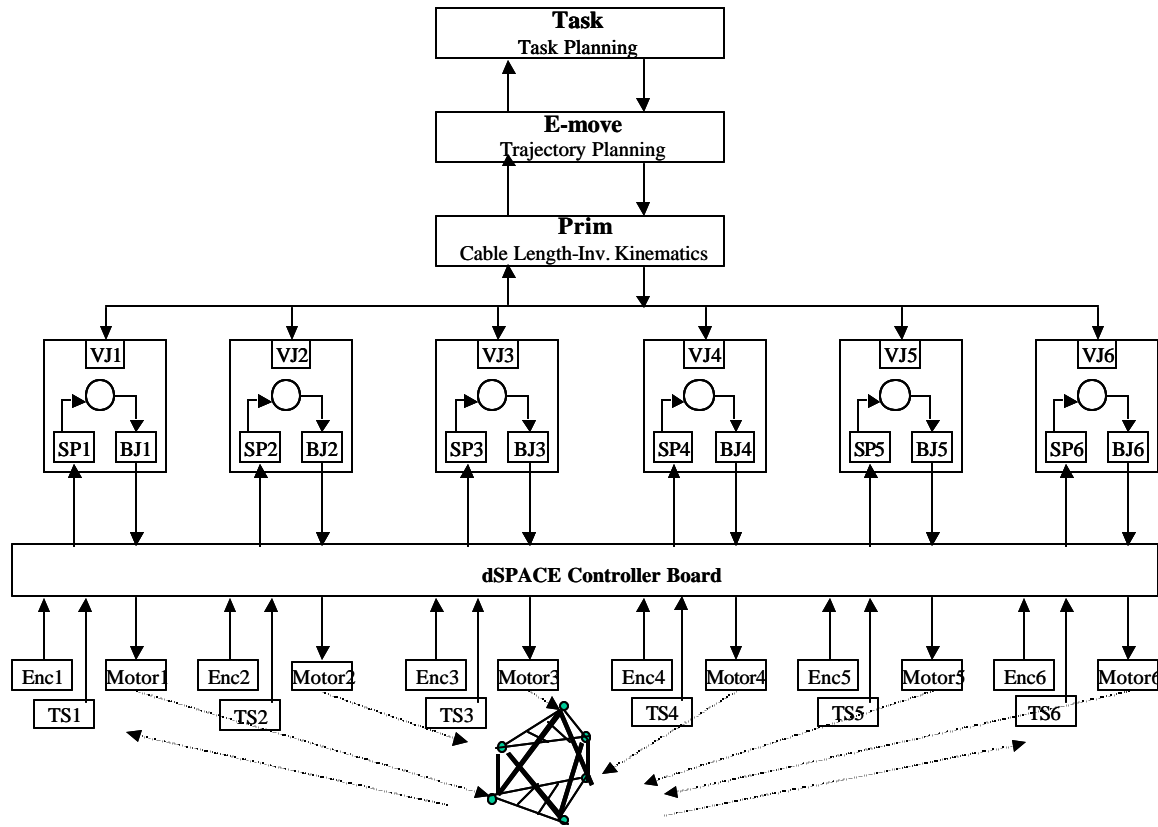


Figure 9. Hierarchical Decomposition of Intelligent Control Design for Cable Robot

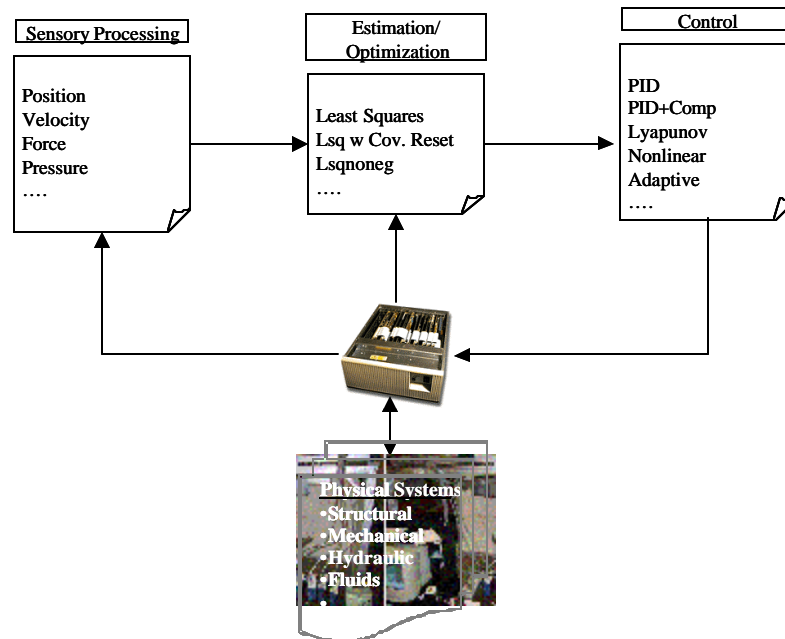


Figure 10. Real-time Estimation, Control, Optimization and System Model Software

